[예제 11-1] ex11-01.c

```
#include <unistd.h>
#include <sys/types.h>
#include <fcntl.h>

#define MSGSIZE 100

main()
{
    pid_t   pid1, pid2;
    int     filedes, p1[2], p2[2];
    fd_set  initset, newset;
    int     nread;
    char    msg[MSGSIZE];

    pipe(p1);
    pipe(p2);

    pid1 = pid2 = 0;
    pid1 = fork();
    if(pid1 > 0)
        pid2 = fork();

    if(pid1 > 0 && pid2 > 0) {                  /* parent */
        printf("[parent] hello!\n");
        close(p1[1]);
        close(p2[1]);

        FD_ZERO(&initset);
        FD_SET(p1[0], &initset);
        FD_SET(p2[0], &initset);

        newset = initset;
        while(select(p2[0]+1, &newset, NULL, NULL, NULL) > 0) {
                if(FD_ISSET(p1[0], &newset))
                        if(read(p1[0], msg, MSGSIZE) > 0)
                                printf("[parent] %s\n", msg);
                if(FD_ISSET(p2[0], &newset))
                        if(read(p2[0], msg, MSGSIZE) > 0)
```

```
                                printf("[parent] %s\n", msg);
newset = initset;
        }
    }
else if(pid1 == 0 && pid2 == 0) {          /* 1st child */
        printf("[fork1] hello!\n");
        close(p1[0]);
        close(p2[0]);
        close(p2[1]);

        dup2(p1[1], 1);
        execl("ex11-01c", "ex11-01c", (char *)0);
    }
    else if(pid1 > 0 && pid2 == 0) {           /* 2nd child */
        printf("[fork2] hello!\n");
        close(p1[0]);
        close(p1[1]);
        close(p2[0]);
        write(p2[1], "from fork2 via pipe", MSGSIZE);

        mkfifo("./fifo", 0666);
        filedes = open("./fifo", O_RDWR);
        nread = read(filedes, msg, MSGSIZE);
        printf("%s (%d)\n", msg, nread);
        close(filedes);
        unlink("./fifo");
    }
    else
        exit(1);
}
```

[예제 11-1-1] ex11-01c.c

```
#include <unistd.h>
#include <fcntl.h>

#define MSGSIZE 100

main()
{
        int filedes;

        printf("[exec] standard output\n");
        sleep(1);

        filedes = open("./fifo", O_WRONLY);
        write(filedes, "from exec via FIFO", MSGSIZE);
        close(filedes);
}
```

[예제 11-2] ex11-02.c

```
#include <unistd.h>
#include <stdio.h>

#define SIZE    512

main()
{
    char msg[SIZE];
    int filedes[2];
    int i;

    if(pipe(filedes) == -1)
    {
        printf("fail to call pipe()\n");
        exit(1);
    }

    for(i = 0; i < 3; i++)
    {
        printf("input a message\n");
        fgets(msg, SIZE, stdin);
        write(filedes[1], msg, SIZE);
    }

    printf("\n");
    for(i = 0; i < 3; i++)
    {
        read(filedes[0], msg, SIZE);
        printf("%s", msg);
    }
}
```

[예제 11-3] ex11-30.c

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>

#define SIZE    512

main()
{
    char msg[SIZE];
    int filedes[2];

    pid_t pid;

    if(pipe(filedes) == -1)
    {
        printf("fail to call pipe()\n");
        exit(1);
    }

    if((pid = fork()) == -1)
    {
        printf("fail to call fork()\n");
        exit(1);
    }
    else if(pid > 0)
    {
        strcpy(msg, "apple is red.\n");
        write(filedes[1], msg, SIZE);
        printf("[parent] %s\n", msg);
    }
    else
    {
        sleep(1);
        read(filedes[0], msg, SIZE);
        printf("[child] %s\n", msg);
    }
}
```

[예제 11-4] ex11-04.c

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>

#define SIZE    512

main()
{
    char *msg[] = {"apple is red", "banana is yellow", "cherry is red"};
    char buffer[SIZE];
    int filedes[2], nread, i;
    pid_t pid;

    if(pipe(filedes) == -1) {
        printf("fail to call pipe()\n");
        exit(1);
    }

    if((pid = fork()) == -1) {
        printf("fail to call fork()\n");
        exit(1);
    }
    else if(pid > 0) {
        for(i = 0; i < 3; i++) {
            strcpy(buffer, msg[i]);
            write(filedes[1], buffer, SIZE);
        }

        nread = read(filedes[0], buffer, SIZE);
        printf("[parent] %s\n", buffer, nread);

        write(filedes[1], buffer, SIZE);
        printf("[parent] bye!\n");
    }
    else {
        for(i = 0; i < 3; i++) {
            nread = read(filedes[0], buffer, SIZE);
```

```
            printf("[child] %s\n", buffer, nread);
        }
        printf("[child] bye!\n");
    }
}
```

[예제 11-5] ex11-05.c

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>

#define SIZE    512

main()
{
    int filedes[2];
    pid_t pid;

    if(pipe(filedes) == -1)
    {
        printf("fail to call pipe()\n");
        exit(1);
    }

    if((pid = fork()) == -1)
    {
        /* fork() 호출 실패 */
    }
    else if(pid > 0)
    {
        close(filedes[0]);
        /* filedes[1]을 지정하여 파이프에 메시지 쓰기 */
    }
    else
    {
        close(filedes[1]);
        /* filedes[0]을 지정하여 파이프로부터 메시지 읽기 */
    }
}
```

[예제 11-6] ex11-06.c

```
#include <unistd.h>
#include <stdio.h>

#define SIZE    512

main()
{
        char *msg1 = "apple is red";
        char *msg2 = "banana is yellow";
        char buffer[SIZE];

        int filedes[2];
        int nread;

        if(pipe(filedes) == -1)
        {
                printf("fail to call pipe()\n");
                exit(1);
        }

        write(filedes[1], msg1, strlen(msg1) + 1);
        write(filedes[1], msg2, strlne(msg2) + 1);

        nread = read(filedes[0], buffer, SIZE);
        printf("%d, %s\n", nread, buffer);

        nread = read(filedes[0], buffer, SIZE);
        printf("%d, %s\n", nread, buffer);
}
```

[예제 11-7] ex11-07.c

```
#include <unistd.h>
#include <stdio.h>

#define SIZE    512

main()
{
        char *msg1 = "apple is red";
        char *msg2 = "banana is yellow";
        char buffer[SIZE];

        int filedes[2], nread;
        int len1 = strlen(msg1) + 1;
        int len2 = strlen(msg2) + 1;

        if(pipe(filedes) == -1) {
                printf("fail to call pipe()\n");
                exit(1);
        }

        write(filedes[1], msg1, len1);
        write(filedes[1], msg2, len2);

        nread = read(filedes[0], buffer, len1);
        printf("%d, %s\n", nread, buffer);
        nread = read(filedes[0], buffer, len2);
        printf("%d, %s\n", nread, buffer);
}
```

[예제 11-8] ex11-08.c

```
#include <unistd.h>
#include <stdio.h>

#define SIZE    512

main()
{
        char *msg1 = "apple is red";
        char *msg2 = "banana is yellow";
        char buffer[SIZE];

        int filedes[2];
        int nread;

        if(pipe(filedes) == -1)
        {
                printf("fail to call pipe()\n");
                exit(1);
        }

        write(filedes[1], msg1, SIZE);
        write(filedes[1], msg2, SIZE);

        nread = read(filedes[0], buffer, SIZE);
        printf("%d, %s\n", nread, buffer);

        nread = read(filedes[0], buffer, SIZE);
        printf("%d, %s\n", nread, buffer);
}
```

[예제 11-9] ex11-09.c

```
#include <unistd.h>
#include <signal.h>
#include <limits.h>

int nc;
void alarm_action(int);

main()
{
        int filedes[2];
        char msg = 'A';

        struct sigaction act;
        act.sa_handler = alarm_action;
        sigfillset(&(act.sa_mask));

        if(pipe(filedes) == -1) {
                printf("fail to call pipe()\n");
                exit(1);
        }
         printf("PIPE size : %d bytes\n", fpathconf(filedes[1], _PC_PIPE_BUF));
        nc = 0;
        sigaction(SIGALRM, &act, NULL);
        alarm(1);
        while(1) {
                write(filedes[1], &msg, 1);
                nc++;
        }
}
void alarm_action(int signo)
{
        printf("\n\nblocked after %d characters\n", nc);
        exit(1);
}
</예제>
```

[예제 11-10] ex11-10.c

```
#include <sys/time.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <unistd.h>

#define MSGSIZE 16

void parent(int [][]);
int child(int []);

void onerror(char *msg)
{
    printf("%s");
    exit(1);
}

main()
{
    int p1[2], p2[2];
    char msg[MSGSIZE];
    int i;
    pid_t pid1, pid2;
    fd_set initset, newset;

    pid1 = pid2 = 0;

    if(pipe(p1) == -1)
        onerror("fail to call pipe() #1\n");
    if(pipe(p2) == -1)
        onerror("fail to call pipe() #2\n");

    if((pid1 = fork()) == -1)
        onerror("fail to call fork() #1\n");
    if(pid1 > 0)
    if((pid2 = fork()) == -1)
        onerror("fail to call fork() #2\n");

    if(pid1 > 0 && pid2 > 0) {
```

```
        printf("parent: %d\n", getpid());
        close(p1[1]); close(p1[1]);

        FD_ZERO(&initset);
        FD_SET(p1[0], &initset);
        FD_SET(p2[0], &initset);

        newset = initset;
        while(select(p2[0] + 1, &newset, NULL, NULL, NULL) > 0) {
            if(FD_ISSET(p1[0], &newset))
                if(read(p1[0], msg, MSGSIZE) > 0)
                    printf("[parent] %s from child1\n", msg);
            if(FD_ISSET(p2[0], &newset))
                if(read(p2[0], msg, MSGSIZE) > 0)
                    printf("[parent] %s from child2\n", msg);
            newset = initset;
}
    }
    else if(pid1 == 0 && pid2 == 0) {
        printf("child1: %d\n", getpid());
        close(p1[0]); close(p2[0]); close(p2[1]);

        for(i = 0; i < 3; i++) {
            sleep((i + 1) % 4);
            printf("child1: send message %d\n", i);
            write(p1[1], "i'm child1", MSGSIZE);
        }
        printf("child1: bye!\n");
        exit(0);
    }
    else if(pid1 > 0 && pid2 == 0) {
        printf("child2: %d\n", getpid());
        close(p1[0]); close(p1[1]); close(p2[0]);

        for(i = 0; i < 3; i++) {
            sleep((i + 3) % 4);
            printf("child2: send message %d\n", i);
            write(p2[1], "i'm child2", MSGSIZE);
        }
        printf("child2: bye!\n");
```

```
            exit(0);
        }
    }
```

[예제 11-11] ex11-11.c

```
#include <sys/types.h>
#include <unistd.h>

main()
{
    char *msg[3] = {"apple is red\n", "banana is yellow\n", "cherry is red\n"};
   int p[2];
   pid_t pid;
   int cnt;

   if(pipe(p) == -1)
   {
       printf("fail to call pipe()\n");
       exit(1);
   }

   if((pid = fork()) == -1)
   {
       printf("fail to call fork()\n");
       exit(1);
   }
   else if(pid > 0)
   {
       printf("[parent]\n");
       close(p[0]);
       for(cnt = 0; cnt < 3; cnt++)
           write(p[1], msg[cnt], strlen(msg[cnt]) + 1);
   }
   else
   {
       printf("[child]\n");
       close(p[1]);
       dup2(p[0], 0);
       execlp("wc", "wc", (char *)0);
printf("[child] fail to call execlp()\n");
   }
}
```

[예제 11-12] ex11-12.c

```
#include <sys/types.h>
#include <unistd.h>

main()
{
    int p[2];
    pid_t pid;

    if(pipe(p) == -1)
    {
        printf("fail to call pipe()\n");
        exit(1);
    }

    if((pid = fork()) == -1)
    {
        printf("fail to call fork()\n");
        exit(1);
    }
    else if(pid > 0)
    {
        printf("[parent]\n");
        close(p[0]);
        dup2(p[1], 1);
        execlp("ls", "ls", "-al", (char *)0);
        printf("[parent] fail to call execlp()\n");
    }
    else
    {
        printf("[child]\n");
        close(p[1]);
        dup2(p[0], 0);
        execlp("wc", "wc", (char *)0);
        printf("[child] fail to call execlp()\n");
    }
}
```

[예제 11-13] ex11-13.c

```
#include <fcntl.h>
#include <sys/stat.h>
#include <unistd.h>

#define MSGSIZE 64

main()
{
        char msg[MSGSIZE];
        int filedes;
        int nread, cnt;

        if(mkfifo("./fifo", 0666) == -1)
        {
                printf("fail to call fifo()\n");
                exit(1);
        }

        if((filedes = open("./fifo", O_RDWR)) < 0)
        {
                printf("fail to call fifo()\n");
                exit(1);
        }

        for(cnt = 0; cnt < 3; cnt++)
        {
                if((nread = read(filedes, msg, MSGSIZE)) < 0)
                {
                        printf("fail to call read()\n");
                        exit(1);
                }

                printf("recv: %s\n", msg);
        }
     unlink("./fifo");
}
```

[예제 11-14] ex11-14.c

```
#include <fcntl.h>
#include <sys/stat.h>
#include <unistd.h>

#define MSGSIZE 64

main()
{
        char msg[MSGSIZE];
        int filedes;
        int cnt;

        if((filedes = open("./fifo", O_WRONLY)) < 0)
        {
                printf("fail to call open()\n");
                exit(1);
        }

        for(cnt = 0; cnt < 3; cnt++)
        {
                printf("input a message: ");
                scanf("%s", msg);

                if(write(filedes, msg, MSGSIZE) == -1)
                {
                        printf("fail to call write()\n");
                        exit(1);
                }
                sleep(1);
        }
}
```